Amendments to the Specification

Please replace paragraph [0001] with the following replacement paragraph:

[0001] The present invention relates [[Lo]] to public key cryptosystems and more particularly to key generation within such systems.

Please replace paragraph [0033] with the following replacement paragraph:

[0033] Referring, therefore, to FIG. 2, a first method of generating a key, k, originates by obtaining a seed value (SV) from the random number generator 26. For the purposes of an example, it will be assumed that the cryptographic function is performed over a group of order q, where q is a prime represented as a bit string of predetermined length [[1]] L. By way of example only it will be assumed that the length [[l]] L is 160 bits, although, of course, other orders of the field may be used.

Please replace paragraph [0034] with the following replacement paragraph:

[0034] To provide a value of k of the appropriate order, the hash function 28 has an [[1]] L bit output, e.g. a 160 bit output. The bit string generated by the random number generator 26 is greater than [[1]] L bits and is therefore hashed by the function 28 to produce an output H(seed) of [[1]] L bits.

Please replace paragraph [0044] with the following replacement paragraph:

[0044] A further embodiment is shown in FIG. 5 which is similar to that of FIG. 4. In the embodiment of FIG. 5, the selection of the required [[1]] L bit string is obtained by applying a [[1-bit]] L-bit wide masking window to the combined bit string.

Please replace paragraph [0045] with the following replacement paragraph:

[0045] This is tested against the value of q and if acceptable is used as the value of k. If it is not acceptable it is rejected and the [[1]] L bit window incremented along the combined bit string to obtain anew value.

Please replace paragraph [0047] with the following replacement paragraph:

[0047] A similar procedure may be used directly on an extended output of the hash function 28 as shown in FIG. 6 by applying a window to obtain the required [[I]] L bit string. The bit string is tested against q and the window incremented until a satisfactory value of k is obtained.

Please replace paragraph [0048] with the following replacement paragraph:

[0048] As shown in FIG. 7, the value of k may be generated by utilizing a low Hamming weight integer obtained by ~~combine~~ combining the output of the random number generator 26 to facilitate computation of an intermediate public key α^k. The integer is masked by combination with predetermined precomputed value k' to obtain the requisite Hamming weight for security. Such a procedure is disclosed in copending Canadian application 2,217,925. This procedure is modified to generate the low Hamming weight integer k as a bit string greater than [[1]] L, for example a 180 bit string The masking value k' is distributed throughout the 180 bit string and the resultant value reduced mod q to obtain a 163 bit value k". Note that the value $\alpha^{k"}$ can be efficiently computed by combining the precomputed value [[α']] α with the efficiently computable value α^k.

Please replace paragraph [0049] with the following replacement paragraph:

[0049] A similar technique may be used by relying on multiplicative masking. In this embodiment the value of k is combined with a value β where $\beta=\alpha^u$. The value of u is a secret value that is used to mask the low Hamming weight of k. Again, the values of u and the low Hamming weight number k can be chosen to have bit lengths greater than [[1]] L, for example, bit lengths of 180. The resultant value is $k"=u^k \bmod q$. It will be appreciated that $\alpha^{k"}$ can be

efficiently computed since $\beta = \alpha^n$ is precomputed, and since k has low Hamming weight.